1.   An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Kuiran Liu, Reg. No. 60,039 on March 14, 2008.

2.   The application has been amended as follows:

<u>In claims</u>

1. (Currently Amended) A system to provide conversation states, comprising:

    a first computing device capable of:

        running a process on the first computing device; and

        accepting a message during a conversation between the process running on the first computing device <u>in a cluster </u>and another process <u>that is running on a computing device outside the cluster</u>, wherein the message is associated with a conversation state that operates to be invoked over an application service ~~such as~~ <u>wherein the application service is</u> a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA);

    a second computing device <u>in the cluster </u>capable of:

        maintaining [[a]] the conversation state requested by the message; and

        storing information of the conversation state in memory on the second computing device; and

    a conversation manager capable of:

        identifying the location of the second computing device which maintains the conversation state requested by the message; and

Attorney's Docket No.: BEAS-01333US1   1
Application No.: 10/733,599
M:\tliu\wp\BEAS\1333US1\beas_1333us1_resp.doc

providing the location and/or the information of the conversation state to the first computing device.

2. (Previously presented) The system according to claim 1, wherein:

the first and second computing devices form a cluster.

3. (Previously presented) The system according to claim 1, wherein:

the conversation manager is capable of maintaining the locations of all conversation states in the system.

4. (Previously presented) The system according to claim 1, wherein:

the information may include, a map of every conversation state leased, owned, or stored on the second computing device.

5. (Previously presented) The system according to claim 1, wherein:

the first and second computing devices are the same computing device.

6. (Previously presented) The system according to claim 1, wherein:

the second computing device is capable of maintaining the information both in-memory and on persistent storage.

7. (Previously presented) The system according to claim 1, wherein:

the conversation manager is capable of designating the second computing device as the primary and replicating the information on the second computing device to a third computing device.

8. (Previously presented) The system according to claim 7, wherein:

the conversation manager is capable of routing to the third computing device and setting it as the new primary when the second computing device fails.

9. (Previously presented) The system according to claim 1, wherein:

the conversation manager is capable of periodically determining the availability of the second and third computing devices.

10. (Currently Amended)   A system to provide conversation for Web service, comprising:

a conversation partner, which is a process, capable of providing a message during a conversation between the conversation partner and a process running on a first computing device;

said first computing device capable of accepting a message during the conversation with the conversation partner, wherein the message is associated with a conversation state that operates to be invoked over an application service such as wherein the application service is a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA);

a second computing device capable of:

maintaining [[a]] the conversation state requested by the message; and

storing information of the conversation state in memory on the second computing device; and

a conversation manager capable of:

identifying the location of the second computing device which maintains the conversation state requested by the message; and

providing the location and/or the information of the conversation state to the first computing device.

11. (Original) The system according to claim 10, wherein:

the message includes a conversation ID.

12-19. (Canceled).

Attorney's Docket No.: BEAS-01333US1   3
Application No.: 10/733,599
M:\tliu\wp\BEAS\1333US1\beas_1333us1_resp.doc

20. (Previously presented) The system according to claim 11, wherein:

the first computing device is capable of contacting the conversation manager to determine the location of the conversation_state requested by the message using the conversation ID.

21. (Previously presented) The system according to claim 10, wherein:

the first computing device is capable of answering a request for the conversation

state directly without contacting the conversation manager if it owns such state.

22. (Previously presented) The system according to claim 10, wherein:

the conversation manager is capable of accepting a request for the location of the conversation_state from the first computing device.

23. (Previously presented) The system according to claim 11, wherein:

the conversation manager is capable of providing the location and/or the information of the conversation_state to the first computing device requesting it based on the conversation ID.

24. (Previously presented) The system according to claim 10, wherein:

the first computing device is capable of accepting the location of the conversation state from the conversation manager.

25. (Previously presented) The system according to claim 10, wherein:

the first computing device is capable of invoking the conversation_state on the second computing device in order to respond to the conversation message received.

26. (Previously presented) The system according to claim 10, wherein:

the conversation manager is capable of sharing the conversation_state with at

least two conversations.

27. (Previously presented) The system according to claim 10, wherein:

the conversation manager is capable of tracking a participating Web service that initiates the conversation.

28. (Previously presented) The system according to claim 27, wherein:

the conversation manager is capable of sharing the conversation_state with at least two Web services and joining the sessions of these services.

29. (Currently Amended) A method to provide a conversation for a Web service, comprising:

maintaining a conversation state on a <u>second</u> computing device <u>in a cluster</u>;

storing information of the conversation state in memory on the <u>second</u> computing device;

accepting a message requesting the conversation state during a conversation between ~~two processes~~ <u>a process running on a first computing device in a cluster and another process that is running on a computing device outside the cluster</u>;

contacting a conversation manager to determine the location of the conversation state requested by the message;

accepting the location and/or the information of the conversation state from the conversation manager; and

invoking the conversation state over an application service ~~such as~~ <u>wherein the application service is </u>a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message.

30. (Currently Amended) A method to provide a conversation for a Web service, comprising:

maintaining a conversation state on a <u>second</u> computing device in a cluster;

storing information of the conversation state in memory on the <u>second</u> computing device;

accepting a message requesting the conversation state during a conversation between ~~two processes~~ a process running on a first computing device in a cluster and another process that is running on a computing device outside the cluster; and

invoking the conversation state over an application service ~~such as~~ wherein the application service is a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message received directly at the computing device without contacting a conversation manager.

31. (Previously presented) The method according to claim 29, further comprising:

maintaining the locations of all conversation states in the system on the conversation manager.

32. (Previously presented) The method according to claim 29, further comprising:

maintaining on the computing device its conversation state information, which may include, a map of every state leased, owned, or stored on it.

33. (Canceled).

34. (Previously presented) The method according to claim 32, further comprising:

maintaining the conversation_state information on the computing device both in-memory and on persistent storage.

35. (Previously presented) The method according to claim 32, further comprising:

designating the computing device as the primary and replicating the conversation state information on the computing device to another computing device.

36. (Previously presented) The method according to claim 35, further comprising:

routing to the another computing device; and

setting it as the new primary when the current primary computing device fails.

37. (Previously presented) The method according to claim 29, further comprising:

determining the availability of the computing devices periodically.

38. (Canceled).

39. (Previously presented) The method according to claim 29, further comprising:

accepting request for the location of the conversation state from a computing device; and

providing the location of the conversation_state to the computing device requesting it.

40. (Previously presented) The method according to claim 29, further comprising:

sharing the state with at least two conversations.

41. (Previously presented) The method according to claim 29, further comprising:

tracking a participating Web service that initiates the conversation.

42. (Previously presented) The method according to claim 41, further comprising:

sharing the conversation_state with at least two Web services; and

joining the sessions of these services.

43. (Currently Amended) A machine readable ~~storage~~ medium having instructions stored thereon that when executed by a processor cause a system to:

maintain a conversation state on a second computing device in a cluster;

store the information of the conversation state in memory on the second computing device;

accept a message requesting the state during a conversation between ~~two processes~~ a process running on a first computing device in a cluster and another process that is running on a computing device outside the cluster;

contact a conversation manager to determine the location of the conversation state requested by the message;

accept the location and/or the information of the conversation state from the conversation manager; and

invoke the conversation state over an application service ~~such as~~ wherein the application service is a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message.

44. (Currently Amended) A machine readable ~~storage~~ medium having instructions stored thereon that when executed by a processor cause a system to:

maintain a conversation state on a second computing device in a cluster;

store information of the conversation state in memory on the second computing device;

accept a message requesting the conversation state during a conversation between ~~two processes~~ a process running on a first computing device in a cluster and another process that is running on a computing device outside the cluster; and

invoke the conversation state over an application service ~~such as~~ <u>wherein the</u> <u>application service is</u> a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message received directly at the computing device without contacting a conversation manager.

45. (Previously presented) The machine readable <u>storage</u> medium of claim 43, further comprising instructions that when executed cause the system to:

maintain the locations of all conversation_states in the system on the conversation manager.

46. (Previously presented) The machine readable <u>storage</u> medium of claim 43, further comprising instructions that when executed cause the system to:

maintain on the computing device information, which may include, a map of every conversation_state leased, owned, or stored on it.

47. (Canceled).

48. (Previously presented) The machine readable <u>storage</u> medium of claim 46, further comprising instructions that when executed cause the system to:

maintain the conversation_state information on the computing device both in-memory and on persistent storage.

49. (Previously presented) The machine readable <u>storage</u> medium of claim 48, further comprising instructions that when executed cause the system to:

designating the computing device as the primary and replicating the conversation state information on the computing device to another computing device.

50. (Previously presented) The machine readable <u>storage</u> medium of claim 49, further comprising instructions that when executed cause the system to:

     route to the another computing device; and

     set it as the new primary when the current primary computing device fails.

51. (Previously presented) The machine readable <u>storage</u> medium of claim 43, further comprising instructions that when executed cause the system to:

     check for the availability of the computing devices periodically.

52. (Canceled).

53. (Previously presented) The machine readable <u>storage</u> medium of claim 43, further comprising instructions that when executed cause the system to:

     accept request for the location of the state from a computing device; and

     provide the location of the state to the computing device requesting it.

54. (Previously presented) The machine readable <u>storage</u> medium of claim 43, further comprising instructions that when executed cause the system to:

     share the state with at least two conversations.

55. (Previously presented) The machine readable <u>storage</u> medium of claim 43, further comprising instructions that when executed cause the system to:

     track a participating Web service that initiates the conversation.

56. (Previously presented) The machine readable <u>storage</u> medium of claim 55, further comprising instructions that when executed cause the system to:

     share the conversation state with at least two Web services; and

Attorney's Docket No.: BEAS-01333US1   10
Application No.: 10/733,599
M:\tliu\wp\BEAS\1333US1\beas_1333us1_resp.doc

join the sessions of these services.

57. (Currently Amended) A system for handling conversation, comprising:

means for maintaining a conversation state on a <u>second</u> computing device <u>in a cluster</u>;

means for storing information of the conversation state in memory on <u>second</u> the computing device;

means for accepting a message requesting the conversation state during a conversation between ~~two processes~~ <u>a process running on a first computing device in a cluster and another process that is running on a computing device outside the cluster</u>;

means for contacting a conversation manager to determine the location of the conversation state requested by the message;

means for accepting the location and/or the information of the conversation state from the conversation manager; and

means for invoking the conversation state over an application service ~~such as~~ <u>wherein the application service is</u> a Java Remote Method Invocation (RMI) or a Common Object Request Broker Architecture (CORBA) on the computing device in order to respond to the conversation message.

58. (Canceled).

59. (Previously presented) The system according to claim 1, wherein:

the conversation can be within the context of a business application.

60. (Previously presented) The system according to claim 1, wherein:

the conversation state can be one of: a program, an application, a service, and a database instance.

3.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Jungwon Chang whose telephone number is 571-272-3960. The examiner can normally be reached on 6:30-2:00 (Monday-Friday).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nathan J. Flynn can be reached on 571-272-1915. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jungwon Chang/
Primary Examiner, Art Unit 2154
March 14, 2008

Attorney's Docket No.: BEAS-01333US1   12
Application No.: 10/733,599
M:\tliu\wp\BEAS\1333US1\beas_1333us1_resp.doc

Application Number	Application/Control No.	Applicant(s)/Patent under Reexamination	
	10/733,599	FRY, CHRIS	
	Examiner	**Art Unit**	
	Jungwon Chang	2154	